Image Classification

Course Name: Machine Learning
Instructor's Name: Vinayak Elangovan
Date: 5/17/2017

Elif Ilaria Yurtseven · Nikesh Yadav · Deepak Shah

Table of Contents

Abstract

Image Classification is a vast sub domain of Computer vision with a variety of applications. In our project, our purpose was to be able to classify a various category of images according to various machine learning algorithms such that after we train the models, the algorithm can differentiate within the difference categories. Our main problem in this project was being able to decide how to reduce the overwhelming number of pixels associated with the images such that we can maintain the important information of the images in order to be able to classify their diversities. In order to do the reduction we used PCA so that the dimensions of the images were reduced and then used algorithms such as SBM, kNN and Random Forest to build models and compare which ones were more appropriate to test image classification with the kind of data that we had. In general, our models classified the images we trained to a great accuracy. We compared all the models and we found that amongst the three algorithms that we used, SVM was the best model with a misclassification rate of only 12%.

Introduction

How many times a day do we search something on google images and get exactly what we want without even realizing what is the work done behind finding all images of a specific object? Image recognition is a sub-domain of Computer vision, a common process vastly used in a variety of fields today. It is part of an interdisciplinary field that deals with how computers can be made to gain high-level understanding of digital images or videos. In 2016, Facebook rolled out an accessibility feature for its iOS app "Automatic Alternative (alt) Text". Automatic alt text provides visually impaired and blind people with a text description of a photo using image recognition technology. For example, if a blind person is browsing Facebook on his phone and he comes across a picture where his beloved son is holding his newly born cute baby, the screen reader will say something along the lines of "this photo may contain: baby, life, family, cuteness etc." This is an example of how useful image recognition can be. An application like automatic alt text took many years of research even for a company as developed as Facebook which shows us how obviously vast and complicated the image recognition field is. Another use of image classification is autonomous cars in order for them to identify shapes and colors and textures of things that are on the road through image recognition in the video feed and make decisions. Similarly, image classification can be also used for video surveillance and security purposes. For the purpose of our project, we are classifying images of five categories: Car, Tree, Waterfall, Beach and Mountain that are resized and transformed to grayscale. We have used some of the data as training using the models we built with after preprocessing the images with PCA, and ultimately tried to compare the testing results of SVM, kNN and Random Forest to see which would give us a lower rate of misclassification for our testing data.

Published Research Papers

Source:
Geitgey, A. (2016, June 13). Machine Learning is Fun! Part 3: Deep Learning and Convolutional Neural Networks.
https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721

In this article, Adam Geigey talks about two popular algorithms used in Image Recognition- Deep Learning and Convolutional Neural Networks. He first starts by saying that every image is actually a sequence of numbers. An image of “8” is just a bunch of numbers. To have an algorithm recognise an image, we just need to observe the numbers. However if the image is not centrally located, then he introduces the idea of searching with a sliding window with search for the “8” throughout the image. This idea is further polished by having different shapes of “8”. A new training data to recognise all types of “8” is built. However there is a better solution with Convolution to recognise an “8” at any where in the image. Convolution is breaking the image into several overlapping image tiles and feeding each image in the neural network.

Source:
Image Recognition | TensorFlow. (n.d.).
https://www.tensorflow.org/tutorials/image_recognition

Although this is not a paper, this is a great resource to understand about Image Recognition licensed by Google. Tenser Flow is a website with many helpful links and in particular, Image Recognition that explains about deep convolutional neural network. Researchers both internal and external to Google have published papers describing all these models but the results are still hard to reproduce. They now taking the next step by releasing code for running image recognition on the latest model, Inception-v3. Inception-v3 is trained for the ImageNet Large Visual Recognition Challenge using the data from 2012. This is a standard task in computer vision, where models try to classify entire images into 1000 classes, like "Zebra", "Dalmatian", and "Dishwasher". For example, here are the results from AlexNet classifying some images:

Since we in our project are planning to classify images, this website is going to be a great resource.

Source:
M. BISHOP, C. (n.d.). Neural Networks for Pattern Recognition
http://cs.du.edu/~mitchell/mario_books/Neural_Networks_for_Pattern_Recognition_-_Christopher_Bishop.pdf

This is a great paper that explains about neural networks in depth. From the perspective of pattern recognition, neural networks can be regarded as an extension of the many conventional techniques which have been developed over several decades. The first chapter is particularly going to be helpful for our project because it explains pattern recognition. The first chapter provides an introduction to the principal concepts of pattern recognition. By drawing an analogy with the problem of polynomial curve fitting, it introduces many of the central ideas, such as parameter optimization, generalization and model complexity. This chapter also gives an overview of the formalism statistical pattern recognition, including probabilities, decision criteria and Bayes' theorem. Chapter 2 deals with the problem of modelling the probability distribution of a set of data, and reviews conventional parametric and non-parametric methods, as well as discussing more recent techniques based on mixture distributions. Aside from being of considerable practical importance in their own right, the concepts of probability density estimation are relevant to many aspects of neural computing. Neural networks having a single layer of adaptive weights are introduced in Chapter 3.

Source:
Sandeep Kumar , Zeeshan Khan , Anurag Jain. A Review of Content Based Image Classification using Machine Learning Approach
http://search.proquest.com/advancedtechaerospace/docview/1198000536/fulltext/A826FFF0DA6945C2PQ/1?accountid=351

Content based image classification is aimed at efficient classification of relevant images from large image databases based on automatically derived imagery features such as shape, texture and color. This is done by dividing the image into categories and labeling each category as a class. For machines it is very difficult to perceive conceptual and physical imagery when differences occur that can make images less clear. The images need to first be preprocessed where they are grouped into meaningful categories. This kind of machine learning methods require an intensive learning and training stages in comparison to non-parametric Nearest Neighbor based image classifiers. Feature extraction of images are by differentiating between visual and semantic content of the images. The features represent general visual content, Histogram is an important method to extract color from images however we are not using such a feature because we are using the images converted to grayscale. The machine learning approach to use support vector machines would be make a decision tree because they combine the predictions of several methods. Ensemble methods based on decision trees are a good starting point for designing a generic system for image classification. Randomized tree are grown by selecting at each node the test attributes which gives a recursive function. The images are propagated into trees and assigned to classes given by trees. Support vector machines, which is the algorithm that we used for the purpose of this project, implement a learning bias and implement hyperplanes. The problem with this is that SVMs generally solve basic two category classification problems while in this project, we have 5. This can be made easier by allowing an error in each dimension. Finally, KNN classification, which we used in this project, use Bayes optimal classifier but cannot generalize much beyond the labeled sets. This method is least efficient when there are only few labelled images for classes with large variability in object shape and appearance.

Source:
Abhishek Pandey, Anjna Jayant Deen, Rajeev Pandey, Content based Structural Recognition for Image Classification using PSO Technique and SVM
http://search.proquest.com/advancedtechaerospace/docview/1499062680/A826FFF0DA6945C2PQ/4?accountid=351

PSO is particle swarm optimization which works with high dimensional datasets and mixed attribute data and determines the performance of image classification after structural recognition based on content of image and comparing the obtained results with those reported for various other classification approaches. The major image feature is the structure of the object contained in the image as outline or contour. Shape representations could be in terms of boundary-based and region-based. The shape of an object is a binary image representing the extent of objects. In region based we consider the shape being composed of a set of two dimensional regions while the boundary based representation presents the shape of its outline. PSO, being a population based iterative learning algorithm, give optimum solution parameters especially with noisy images. PSO consists of three steps, evaluating the fitness of each particle, updating individual and global best and updating velocity and position of each particle. SVM is a new neural network technique that is a linear and bilinear classifier. The proposed PSO-SVM technique would be a fast texture extraction technique that consists of first optimizing the accuracy of SVM and then searching the optimal particle iteratively and the optimal feature subset combinations using PSO. First we would read the image, then convert it into grayscale, and choose the region of interest, then classify all images with region-based classification. According to this article, PSO-SVM is the optimal method that can be used for image classification.

Design Challenges and Design Development

Objectives

Our objective for this project is to categorize 200 images into 5 categories: cars, beaches, waterfalls, trees and mountains and build a model (write a program) using more than one machine learning algorithms that can recognize the differences in these images and after trained, can tell us which new image belongs to which one of the 5 categories we have defined. In order to build the model, we will resize each image to the same size and extract the grayscale pixels of each image in order to use Principal Component Analysis and reduce the dimension of those pixels. Then we will use Support Vector Machine on the reduced set of pixels to build a model, and we will use that model to predict the categories of the new images.

Specifications

The problem at hand is being able to correctly classify what category (out of certain number of categories) an image belongs to. We are inputting a certain number of images for each category and we are expecting the output to be the prediction of which category a new image (an image that was not part of the training data) belongs to. Since this is a supervised learning problem, we need to know the classes of all the images in each category. Then we can use that information as a training data upon which to build a model.

Challenges

We constantly revised our model to reduce the rate of misclassification in the testing data. Some of the challenges were:

1. Finding images that were similar enough so that the algorithm can classify the training data but different enough so that we can test images that weren't already in the training data for each category

2. We initially had ‘house’ as one of our categories but we had difficulties having a low misclassification rate when we had house as a category as there are too many variations of shapes and sizes for a house
3. Number of columns to choose when applying PCA gave different misclassification rate

At the early stage of our project, we searched images by keywords such as “tree”, “car”, etc and not all of the results in google search were relevant images. At the same time, because we are working on a relatively small scale of images, we had to make sure that the images we choose are relatively similar. In addition, some waterfalls had great height and others were relatively very small compared to the size of the image. Hence, we can safely say that our data was not very consistent. As a result we had a high misclassification rate of almost 60% which was technically better than random guessing (which would have been %80) but still not at all close to the number we aimed to achieve. To overcome this problem, we searched for visually similar images using “google image” search. We narrowed our search to selecting images of the same dimension, similar content and similar features (such as shape and dimension of the object of interest compared to the dimension of the entire image. This is how we solved challenge number one on the above list.

Next challenge was the houses. Since houses are of different shapes and we do not use color as a feature in our image classification, the pixel content in the images were dissimilar to each other. Compared to other images of waterfall and tree, our misclassification in houses were a lot higher. After noticing the pattern, we decided to remove “houses” from the categories we were using in our data. We instead added a new category “mountain.” When we were looking for mountain images, we realized that it was much easier to find images similar to each other, as most were shining white and had rugged stone shape. This way, we reinforced consistency in our image categories and solved the challenge number two from the above list.

When we performed Principal Component Analysis (PCA) for the images that were initially 800*600 pixels and decided to reduce them we did not know what would have been the best number of columns. We knew that we were going to have 200 rows because we had a total of 200 images but we had 480000 columns that we wanted to reduce by still keeping the most important pieces of information for each image. We finally decided to reduce the images to 200*200 and hence we ended up with a matrix that had 200 rows and 40000 columns. Our model started with choosing the 200 most important columns. So our data frame was reduced to 200 by 200. However we later realized that choosing the 150 most important columns reduced the error by even more. So we updated our model by choosing the 150 most important columns ending up with a matrix of 200*150. Thus we were able to overcome all challenges one by one.

Implementation and Results

We collected 40 images with dimension 800 * 600 per category. We converted each of those images to grayscale and resized them to 200 * 200.

For example, here is an image of a tree of size 800 * 600.

Once we convert this image to grayscale and resize it to 200 * 200, below is the resulting image looks.

In order to perform this operation we used the following R handy-dandy code.

```
gray_pixels <- function(Image) {
  im <- load.image(Image)
  gray <- grayscale(im)
  resize.im <- resize(gray, w=200, h=200)
  mat <- matrix(resize.im, nrow=1)
  return(mat)
}
```

This code returns a matrix with 1 row of all the resulting pixels. We did this so that we could use this function to extract a row of pixels of each image and stack them together to create a data frame.

With this operation, we reduced 480,000 pixels to 40,000. What we were trying to do was to build a model using all the pixels in the image. However, 40,000 predictors for the response variable was still a lot. Hence, we used Principal Component Analysis to reduce the dimensions of the 200*40000 matrix we have obtained. PCA resulted in 200 components but as we found out that the first 150 components have a cumulative variance proportion of 98%, we decided to take only the first 150 principal components out of the 200. 98% of the variance in the entire data is explained by those 150 components.

Initially we had 200 images with 480,000 predictors. After we reduced our 200 * 480,001 dimensional data to 200 * 151 using PCA, we were able to build a model. At that point, we split the data as 75% for training and 25% for testing. We then used Support Vector Machine using radial kernel where we needed to specify the parameters “cost” and “gamma”. To find the best value of these parameters, we performed a cross validation and picked out the values that corresponded to least error rate. By doing so, we got the value of cost to be 1.265 and that of the gamma to be 0.0001. Then, we performed prediction on our testing data and below is a confusion matrix for the classification.

Prediction\Test	Beach	Car	Mountain	Tree	Waterfall
Beach	11	0	1	0	1
Car	0	7	0	1	0
Mountain	0	0	7	1	0
Tree	0	1	0	8	1
Waterfall	0	0	0	0	10

The misclassification error rate for this confusion matrix was 12%. For the sake of comparison between various machine learning algorithms, we also performed k-Nearest Neighbor and

Random Forest to find the misclassification error rate. Random Forest is an algorithm that constructs a multitude of decision trees, in our case the algorithm built 500 decision trees at training time and outputted the class that is the mode of classification or regression of individual trees. Decision trees were briefly explained in the summary of Abhishek Pandey's article on Image Classification. Random Forest gave us a misclassification rate of 25%. As explained in the same article, kNN stores all available cases and classifies based on a similarity measure (for example: distance functions) for statistical estimation and pattern recognition. Such algorithm would have been more useful to us if we were able to use OpenCV and perform image segmentation so that we could have used exactly the pieces of the images that were of our interest. Unfortunately, since we did not have the time and resources to perform such operation, kNN gave us a misclassification rate of 60%. From our results, we can clearly see that SVM outperformed all other algorithms. We did not necessarily, explain Random Forest and kNN in detail because there was a very easy way to perform these two operation in Rstudio using the language R.

Conclusion

We have come to learn and appreciate a lot about image classification and its application in various fields. We have learnt that it's not a menial task to try and classify images because there are several challenges that are interesting to try and solve. All objects have different shapes, sizes and features and sometimes if it difficult to extract those features and train the models. We were fortunate enough to reduce the error rate to 12% because we confined ourself to a limited class of images that were highly distinguishable from each other and were concentrated on on specific shape as object of interest. We only used 5 classes of images namely Tree, Beach, Waterfall, Mountain and Car. We also made sure that the images we were using were somewhat similar so that our model could pick up the specific features of specific categories for the training set of our data. At first, we were only using 20 images per category and training the model with 75% of the total images but we were getting high misclassification rate of more than 50%. However, when we increased the number of images to 40 per category, even if we kept the training data percentage as 75% of total images, we managed to significantly reduce the error to less than 15% on average. Hence, we have learnt that increasing the images in training set helps the model pick out features more easily and it becomes more accurate with its category prediction of the testing data. Along with this, we noticed that certain algorithms work better than others in image classification. Support Vector Machine proved to be the best algorithm among the ones we used. k-NN and Random Forest did not perform well. Overall, we all felt this was a very interesting project to work on and now we have basic skills and expertise to do further research in image classification in the future.

Team Member’s Contribution

Despite being a small team of 3 people, we divided our work proportionately and each of us provided equal contribution in the project. All of us chose the list of tasks that were most suitable to us. We collaborated together sharing the updated codes after either of us had made some changes.

Elif

1. Searching all the images of mountain, beach, car, tree, houses and waterfall
2. Reading research papers on different methods used for image classification and summarizing them
3. Working together with Nikesh on Principal Component Analysis

Nikesh

1. Implementing the Random Forest Algorithm
2. Implementing the Support Vector Machine
3. Preparing the presentation slides

Deepak

1. Reading research papers on Image Classification and summarizing them
2. Implementing the k-Nearest Neighbour Algorithm and checking if it gives a lower misclassification rate
3. Preparing the presentation slides and adding “art” to visualize in slides

Although there were areas that individually focused on, we met quite often updating each other about the project’s status. We then came together to prepare our report and discuss about the techniques to improvise the project. We all came to know each other’s strengths and weaknesses and in the future we plan to collaborate on projects together.

References

1. Blog, Guest, Sunil Ray, and Kunal Jain. "Best Way to Learn KNN Algorithm in R Programming." *Analytics Vidhya*. N.p., 24 Sept. 2015.

2. Geitgey, Adam. "Machine Learning Is Fun! Part 3: Deep Learning and Convolutional Neural Networks." *Medium*. N.p., 13 June 2016.

3. Kumar, Sandeep, Khan, Zeeshan, Jain, Anurag. "A Review of Content Based Image Classification using Machine Learning Approach" *International Journal of Advanced Computer Research*. 5 Sept. 2012.

4. Pandey, Abhishek, Deen, Anjna Jayant, Pandey, Rajeev. "Content based Structural Recognition for Image Classification using PSO Technique and SVM" *International Journal of Computer Applications*. 7 Feb. 204.

5. Bishop, Christopher. "Neural Networks for Pattern Recognition"

6. Image Recognition, TensorFlow

7. Google Images